Lecture 14

Functions, Procedures and Classes.

primitive values and objects.

Files.

Mar 4, 2011

Prof. R K Joshi
Computer Science and Engineering
IIT Bombay
Email: rkj@cse.iitb.ac.in
Revision

- **global variables**: functions can share variables
  - but they can be accessed by any function
- **class**: set of member functions + set of private variables + constructor (+ destructor: we will talk about it later)
- **object**: instance of a class
- **class** is like a type and **object** is like a value
- **private vs. public**

- **member functions**
  - usual input parameters
  - usual return value
  - + sharing of private variables
- **dot operator**: for invocation of member functions
- **a return value** can be assigned as usual
- **many instance of a class** can be created
- **values of private variables** define the state of an object at a given point of time
The model of functions: function body vs. function invocation

- **function body:**
  - definitions
  - local variables,
  - accesses to input parameters
  - computing a return value

- **function invocation**
  - actual input parameters
  - actual local variables

- **actual return Values**
While pure functions do not use history, procedures may use:

- procedure body: definitions
- local variables, accesses to input parameters, computing a return value
- global
- actual parameter invocation
- actual local variables
- return Values
Inovcation can access the global variable

procedure body:
- definitions
- local variables,
  accesses to input parameters
  computing a return value
Multiple invocations: separate copy of local variables, but a shared copy of globals

procedure body:

- definitions
- local variables, accesses to input parameters computing a return value

- actual
- input parameters
- local variables
- procedure invocation
- return Values
- actual
- input parameters
- local variables
- procedure invocation
- return Values
- global
- global
Classes
private variables

Private variable
Private variable
Classes

member functions

- Private variable
- Private variable
- Member function
- Member function
Classes
sharing of private variables by member functions
Classes
constructor
Classes
constructor initializes private variables

Diagram:
- Private variable
- Member function
- Constructor
Objects: Instances of classes

Private variable

Member function

Member function

constructor

Object 1

Object 2

Object 3
Objects: Each one keeps a separate copy of the set of private variables.
Objects: The dot operator to **invoke** member functions on objects

*(also called sending messages to objects)*

- Private variable
- Member function
- Member function
- Constructor

Invoke a member function
Supplying input parameters
Objects: a result comes out of member function invocation

Private variable

Member function

Member function

constructor

Invoke a member function
Supplying input parameters

Out comes the return result

Object 1

Private Variable values

Object 2

Private Variable values

Object 3

Private Variable values
The object's state (private variables) may get modified during member function invocation.

Invoke a member function
Supplying input parameters

Out comes the return result

Object 1
Object 2
Object 3

Private Variable values

Private variable
Private variable
Private variable

Member function
Member function
constructor
Libraries of classes

- A pre-implmented collections of classes
- Used for reusability
- Separately compiled and kept somewhere in the system
- The programmer includes the declarations
- The pre-compiled libraries are linked before the executable file gets produced
  - Recall separate compilation
Files

• A file is a unit of storage on disk

• You have different kinds of files on your disk
  – Text files, documents
  – Pictures, photographs
  – Music
  – Source code
  – Executable files
  – Lab handouts
Creating and accessing our own files through C++

- You have so far used editors or copy commands to create your own files
- You can also create them from a program
- Or.. existing files can be accessed and modified from a program
- Scientific, data oriented computations
  - Programs have computations
  - Data comes in from files or goes out into files
File handling library

- Include `<fstream>`
  - For input and output on files
  - Classes
    - For performing output on files
    - For input from files
    - For doing both both
- Class ofstream
- Class ifstream
- Class fstream
Some Member functions of files

- `open()`  
  opening a file

- `is_open()`  
  checking if it's open

- `eof()`  
  end of file reached?

- `close()`  
  closing a file

- `operator <<`  
  for output to file

- `operator >>`  
  input from file